SCIS 2012 The 29th Symposium on
Cryptography and Information Security
Kanazawa, Japan, Jan. 30 - Feb. 2, 2012
The Institute of Electronics,
Information and Communication Engineers

情報漏洩元の特定を可能とする電子文書管理システム

A Document Management System to Specify the Source of Information Leakages

今井 正樹 * 上原 哲太郎 † 侯 書会 ‡ 上田 浩 § 津田 侑 *
Masaki Imai Tetsutaro Uehara Shuhui hou Hiroshi Ueda Yu Tsuda
喜多 一 §
Hajime Kita

あらまし 企業等の組織において情報資産管理をするツールの一つに電子文書管理システムがある．電子文書管理システムは一般にアクセス制御機能を備え，不正アクセス等による情報漏洩を防止できる．しかし，アクセス制御機能では，正当なアクセス権を持つ者による情報漏洩は対策が困難である．そこで，情報漏洩を心理的に抑止できる機能として，文書にユーザを特定可能な ID（以下，ユーザ ID）を電子透かしで埋込む機能が提案されている．この機能は文書に埋込まれたユーザ ID から情報漏洩元が特定可能であるという原理により情報漏洩を抑止する．しかし，従来のユーザ ID 埋込み機能は，結託攻撃と呼ばれる手法により文書に埋込まれたユーザ ID を改竄されるという問題がある．本稿では，文書形式として広く用いられている OOXML（Office Open XML）を対象として，結託耐性符号を用いて符号化したユーザ ID（以下，耐結託 ID）を埋込む電子文書管理システムを提案し，システムの実現可能性を評価する．耐結託 ID は，一般的な ID に比べ ID 長が大きく，従来の電子透かしでは埋込み容量に関して問題があった．提案システムではテキストボックスを利用した電子透かしを用いてこの問題に対処する．実現可能性の評価では，提案システムを用いることで，結託攻撃された文書の耐結託 ID から結託したユーザを少なくとも一人特定可能であること等を示した．

キーワード コンテンツ保護，電子透かし，結託耐性符号，電子文書管理

1 はじめに

企業等の組織の情報資産の中で，文書は大きな地位を占めており，文書の流通管理が問題となっている．文書の流通を管理する方法の一つとして，電子文書管理システムがある．電子文書管理システムは，一般にアクセス制御機能を備え，不正アクセス等による情報漏洩を防止できる．一方，JNSA の報告 [1] によると，正当なアクセス権を持つ者による個人情報漏洩は，件数は少ないものの発生した場合の被害は大きい．また，アクセス制御機能では正当なアクセス権を持つ者の文書入手は防止困難である．

そこで，利用者に情報漏洩に対してリスクを負わせ，情報漏洩を心理的に抑止する方法が考えられる．例えば，文書の流通を追跡可能とする仕組みにより，漏洩行為が発覚するリスクを負わせる手法である．

そのような仕組みを持つ電子文書管理システムの情報漏洩対策として，ログ管理機能がある．ログ管理機能は組織内の PC に専用のソフトウェアを導入する等して文書の流通を記録する．システム管理者による組織内での情報漏洩経路の推定を容易にし，情報漏洩した者の特定を効率化できる．一方，組織外の環境でも効果を持つ機能として，文書にユーザを特定可能な ID（以下，ユーザ ID）を電子透かしで文書に埋込む機能が提案されている．この機能は，文書に埋込まれたユーザ ID から情報漏洩元が特定可能であるという原理により，情報漏洩を抑止する．しかし，従来のユーザ ID 埋込み機能は結託攻撃 [2] と呼ばれる手法により，文書に埋込まれたユーザ ID が改竄される危険性がある．

本稿では，文書形式として，企業等の組織で広く用いられている OOXML（Office Open XML）を対象として，結託耐性符号 [2] により符号化したユーザ ID（以下，**耐結託 ID**）を埋込む機能を備えた電子文書管理システムを提案し，その実現可能性を評価する．本提案により，

* 京都大学 大学院情報学研究科, 〒 606-8186 京都市左京区吉田二本松町. Graduate School of Informatics, Kyoto University, Nihon-matsu, Yoshida, Sakyo, kyoto, 606-8186, Japan.
† NPO 法人情報セキュリティ研究所
‡ 北京科技大学 数力学系
§ 京都大学 学術情報メディアセンター

結託攻撃されたユーザ ID から結託した ID が特定可能となり，電子文書管理システムにおける情報漏洩を抑止する効果が高まる．

2 関連研究

2.1 電子文書管理システムの情報漏洩対策機能

電子文書管理システムは，文書に作成日時，作成者等のメタデータの付与により，文書の検索，文書の共有や流通管理を可能とする．組織における情報漏洩経路には，印刷した紙文書の盗難・紛失，PC や外部記憶装置の盗難・紛失，メールによる文書の誤送信，不正アクセス等がある．電子文書管理システムは，データの暗号化や認証等によるアクセス管理，ログ管理，電子透かしによる属性情報の埋込み等の機能を備え，こうした経路からの情報漏洩に対策している．

電子透かしによるメタデータの埋込み機能を備えた電子文書管理システムとしては，文書印刷時に紙文書にユーザ情報を挿入する秘文[1]や，電子文書に対してユーザ ID の埋込みをする InfoCage[2]がある．

電子文書管理システムは多様な情報漏洩経路に対策している．しかし現状では，例えば文書内容を手書きで書き写した文書，文書をキャプチャした画像，組織外で印刷された文書等による情報漏洩は対策が難しい．情報漏洩が想定される状況ごとに対策を取り，情報漏洩リスクを軽減する必要がある．

2.2 結託耐性符号による情報漏洩抑止

ユーザ ID 埋込み機能は，文書へのユーザ ID 埋込みをユーザに周知することにより情報漏洩の心理的抑止を狙う．したがって，正当なアクセス権を持つ者が情報漏洩を起こす際にはユーザ ID の改竄を試みると想定できる．

電子透かしへの典型的な攻撃手法に結託攻撃がある．結託攻撃とは異なる ID が埋込まれた内容が同一の文書を比較し，データの異なる部分のみを改竄する手法である．従来のユーザ ID 埋込み機能は結託攻撃によりユーザ ID が改竄されるという問題がある．

結託攻撃に対処する方法として結託耐性符号 [2] がある．結託耐性符号のうち追跡性を持つ符号は，ある条件下で攻撃された符号から結託した符号の幾つかを特定できる．条件とは Marking Assumption[2] が成立し，結託攻撃した人数が一定数以下であることである．本稿ではこの一定数を**耐結託数**と呼ぶ．Marking Assumption とは電子透かしで異なる符号を埋込んだ文書を比較した場合に，データの異なる部分のみが改竄されるという仮定である．

この結託耐性符号を利用した耐結託 ID を用いることでユーザ ID 埋込み機能の情報漏洩抑止の効果を向上できる．図 1 は電子文書管理システムからダウンロードした文書に耐結託 ID が埋込まれることで，情報漏洩が抑止される様子を示している．

図 1(1)(2) のように電子文書管理システムにアップロードされた文書をユーザ A，B，C がダウンロードすると，ユーザが持つ文書には，ユーザ ID 埋込み機能によりそれぞれ異なる耐結託 ID が電子透かしで埋込まれる．このとき悪意のあるユーザ B，C が結託攻撃すると，図 1(3)(4) のように文書の耐結託 ID が改竄され，文書をダウンロードしたユーザが不明確になる．ユーザ B，C は耐結託 ID による情報漏洩抑止力が無くなるため，図 1(5) のように改竄した文書をネットワーク上に漏洩する可能性がある．

しかし，漏洩文書がネットワーク上からシステム管理者に検出されると，図 1(6)(7) のようにシステム管理者は結託耐性符号の特徴を利用して改竄前の符号を特定できる．そのため，図 1(8) のようにユーザ B，C には漏洩文書から得られた耐結託 ID により，情報漏洩の容疑がかかる．このような原理から，ユーザ ID が耐結託 ID であることを事前にユーザに周知しておくことで結託攻撃を抑止できる．

2.3 耐結託 ID 埋込み機能に適した結託耐性符号

企業等の組織に属する従業員数は規模の大きな組織では数十万人になる[3]．したがってユーザ ID の符号化に用いる結託耐性符号も数十万の ID を表現する必要がある．

結託耐性符号は，符号長に比例して表現可能な ID 数と耐結託数が大きくなるが，その増加の程度は符号により異なる．追跡性を持つ結託耐性符号のうち，他に比べてこうしたパラメータの増加が大きな符号に ACC 符号（Anti-collusion code）[3] がある．ACC 符号は符号アルファベット以外へ改竄されることを許容しており，符号アルファベットがどのような文字に改竄されるかわからない文書での利用に適している．

Hou らは ACC 符号を改良した 2 つの符号構成法を示している [4]．本稿ではこの 2 つの構成法のうち，小さい符号長でより大きな ID 数が表現可能な unital code を用いた ACC 符号を利用する．表 1 に unital code を用いた ACC 符号の性能を示す．多くの組織で利用可能な ID 数を 20 万個と想定すると，表 1 より，$p = 23$ のと

[1] 株式会社日立ソリューションズ，http://hitachisoft.jp/products/hibun/（参照日付 2011 年 12 月 26 日）

[2] 日本電気株式会社，http://www.nec.co.jp/cced/infocage/index.html（参照日付 2011 年 12 月 26 日）

[3] 日本の大企業として知られるトヨタ，パナソニック，日立は 30 万人規模の従業員数を持つ．
http://www.toyota.co.jp/jpn/company/about_toyota/outline/index.html（参照日付 2011 年 12 月 26 日），
http://panasonic.co.jp/company/info/about/(参照日付 2011 年 12 月 26 日),
http://www.hitachi.co.jp/about/corporate/index.html
（参照日付 2011 年 12 月 26 日）

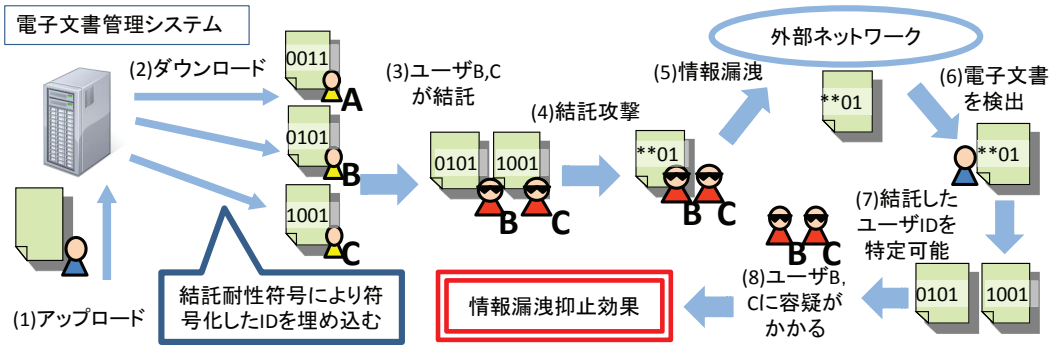

図 1: 結託耐性符号による情報漏洩抑止

き，符号は 268, 203 個の ID を表現可能で必要な性能をみたす．十万人規模の組織でも 1 度は ID の更新が可能である．このとき符号長は 12, 168bit である．したがって耐結託 ID の埋込みに用いる電子透かしは，文書の内容によらず 10^4bit 以上の埋込み容量を持つ必要がある．

2.4 OOXML で利用可能な電子透かし

耐結託 ID を文書へ埋込む際に利用する電子透かしは，結託耐性符号の前提となる Marking Assumption を満たす必要がある．OOXML 文書で利用可能な電子透かしで，Marking Assumption を満たす手法に，改行時の文字数を利用する手法 [5]，文章中の単語を同義語に置換する手法 [6]，空白文字を利用する手法 [7] 等が提案されている．

改行時の文字数を利用する手法は，文章の改行位置を調整し，1 行の文字数や単語内での改行位置等に情報を埋込む．単語を同義語に置換する手法は，文章中の名詞等を予め用意した辞書を用いて埋込みたいビット列に対応する同義語に置換する．空白文字を利用する手法は，Word 文書において空白文字の色が表示に影響しないことを利用し，空白文字の色属性に情報を埋込む．

これらの手法は，文書の文字数に依存して埋込み容量が変化し，比較的埋込み容量が大きいと考えられる空白文字を利用する手法であっても一般的な文書に対する埋込み容量は 10^4 程度である．耐結託 ID の埋込みでは，文書の内容によらず 10^4bit 以上の埋込み容量が必要であり，これらの手法は不適である．

表 1: 結託耐性符号の性能

符号長（bit）	表現可能な ID 数	耐結託数
p^3+1	$p^2(p^2-p+1)$	$p+1$

p は素数を表す

3 OOXML 文書への結託耐性符号の埋込み

耐結託 ID 埋込みには文章の内容によらず，大きな埋込み容量を持つ電子透かしが必要となる．本章では，OOXML の機能の一つであるテキストボックスを利用し，このような条件を満たす電子透かし手法を提案する．

3.1 テキストボックスを利用した電子透かし

OOXML は Microsoft Office 2007 以降で採用されているファイル形式であり，図 2 のような複数の XML ファイルで構成されている．OOXML では図形に文字を挿入するテキストボックスという機能が定義されている．Microsoft Office Word 2007（以下，Word 2007）におけるテキスト入力可能なオブジェクトであるテキストボックスもこの OOXML の同名の機能を用いて実装されている．

本稿では，このテキストボックスを利用して電子透かしを実現する．テキストボックスには任意の文字列を格納できるため，電子透かしに応用する場合，符号の自由度が高い．このことを利用して Marking Assumption を満たすような電子透かしを実現できる．秘匿性についてはテキストボックスを印刷領域であるページの外に配置することで行う．Word 2007 では，ページ外のテキストボックスは画面に表示されず，選択も困難であるため秘匿性が高い．

テキストボックスは，本文やヘッダ等，Word 2007 で表示される文字を扱う部分で宣言することができる．本手法では，本文情報を持つ図 2 の `document.xml` へ任意の文字列を埋込む．

`document.xml` は図 3 のような構造を持つ．root 要素は`<w:document>`であり，`<w:body>`要素以下に文書の内容，フォント等を示す要素を記す．`<w:p>`は文章の段落を表し，`<w:r>`は文を表す．実際の文字は`<w:t>`要素内に書き込む．OOXML 文書では，`<w:p>`要素を繰り返すことで文章を表現している．

テキストボックスは`<w:r>`の子要素であり，図 4 のよ

ファイル名とパス
[Content_Types].xml
docProps/app.xml
docProps/core.xml
customXml/item1.xml
customXml/itemProps1.xml
customXml/rels/item1.xml.rels
word/document.xml
word/fontTable.xml
word/settings.xml
word/styles.xml
word/webSettings.xml
word/document.xml
word/theme/theme1.xml
word/_rels/document.xml.rels
_rels/.rels

図 2: OOXML 文書の構成ファイル

```
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>文章</w:t>
      </w:r>
    </w:p>
    ...
    <w:p>
      <w:r>
        <w:t>文章</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

図 3: document.xml の XML コード

```
<w:pict>
  <v:shape id=“ … ” style=“ … ” >
   <v:textbox >
    <w:txbxContent>
     <w:p>
      <w:r>
       <w:t>文章</w:t>
      </w:r>
     </w:p>
    </w:txbxContent>
   </v:textbox>
  </v:shape>
</w:pict>
```

図 4: テキストボックスの XML コード

うな構造を持つ．図 4<v:shape>は図形を表す要素であり，style 属性で図形の形や位置を定める．テキストボックスは子要素に<w:t>を持ち，文字列を挿入可能である．

本手法では style 属性の図形の位置を示す値を，テキストボックスがページ左外側になるように設定し，<w:t>に任意の文字列を書き込んだテキストボックスを用意する．<w:document>の任意の<w:p>要素以下に用意したテキストボックスを埋込むことで，文書への情報の秘匿を実現する．

Word 2007 では，テキストボックスが埋め込まれた<w:t>要素を含む部分を別文書にコピーした場合，その内容が保持される．よって，本手法は文書ファイルの流出のみならず，文書の一部分を他の文書にコピー&ペーストような場合でも流出元特定に利用できる可能性がある．

3.2 PDF への変換時の電子透かし

企業等の組織で OOXML と同様に広く利用されている文書形式に PDF がある．そこで，本手法で情報を埋込んだ OOXML 文書を PDF ファイルに変換し，変換した PDF ファイルに埋込んだ情報が残っているかを検証する．OOXML 文書の PDF 変換では，1）Word 2007 の PDF/XPS 保存機能を利用する，2）Acrobat PDFMaker Office COM アドインを利用する，3）文書を一旦 PS ファイルに変換して Adobe Distiller を利用するという 3 つの手法を利用した．変換した PDF ファイルから，Adobe Acrobat 9 Pro の書き出し機能を利用して Word 文書を作成し，埋込んだ文字列が表示されるかを検証する．

検証の結果 1），2）の場合，作成した Word 文書には埋込んだ文字列が本文に書き込まれていた．3）の場合に OOXML 文書に埋込んだ文字列が Word 文書に書き込まれていなかったのは，OOXML 文書を PDF ファイルに変換する前に PS ファイルに変換した際に，ページ外に配置された部分について削除されてしまったためであると考えられる．

検証において Word 文書に書き出された文字列は，テキストボックスの枠内に表示された部分のみであった．PDF ファイルに情報を残すためには，提案手法で埋込む情報はテキストボックス枠内に納めておく必要がある．

3.3 提案手法の編集耐性

本手法で埋込んだ情報は Word 2007 を利用した文書操作や，文書を構成する XML ファイルの編集により消去される危険性がある．

Word 2007 を利用した文書の操作では，範囲選択の領域をページ外まで広げて削除する，他の図形を選択中に Tab キーを利用して，テキストボックスを選択して削除する，テキストボックスが属している行をドラッグで選択して削除する等の方法がある．

範囲選択の領域をページ外まで広げる方法については，テキストボックスをより本文より遠い位置に設定することで困難にできる．Tab キーを利用する方法は，文書編集で必ずしも必要な操作ではなく，テキストボックスが削除される危険性は低いと考えられる．テキストボックスが属する行を削除する方法については，文書編集に使われる操作であり実施される危険性がある．埋込んだ情報を完全に削除されないようにするには，文書に複数のテキストボックスを埋込む等して対策する必要がある．文書内に複数のテキストボックスを埋込むと文章がコピーされた場合に，埋込んだ情報が共にコピーされる可能性を高めることもできる．

XML ファイルを直接編集してテキストボックスに埋込まれた情報を削除するには，複数の XML ファイルから document.xml に情報が埋込まれていることを特定し，テキストボックスの<w:t>要素内の情報を削除する必要

がある．しかし，編集方法によっては OOXML 文書が Word 2007 で表示不可能となるため，OOXML に関する知識を持たないユーザは，XML ファイルを編集しての改竄が困難であると考えられる．

3.4　提案手法の埋込み容量に関する検証

提案手法を利用して OOXML 文書に文字列を埋込み，企業等で利用する際に十分大きな ID 数を表現可能な結託耐性符号が埋込み可能であるかを検証した．仕様上，テキストボックス内に書き込める文字数に制限はない．

提案手法を利用して半角で 10^5 字を埋込んだ OOXML 文書を作成した．結果，文書は Word 2007 で表示・保存が可能であった．したがって提案手法は耐結託 ID 埋込み機能に十分な埋込み容量を持つといえる．

4　結託攻撃を抑止する電子文書管理システム

4.1　システム概要

本システムは企業等の組織に属する者が組織内ネットワークから PC 等の端末で利用することを想定した Web アプリケーションである．データベースを用いて OOXML 文書を管理する．

本システムは文書のアップロードやダウンロード，検索等の機能を持つ．情報漏洩対策としてログ管理機能と耐結託 ID 埋込み機能を備え，パスワードを用いた認証機能によりアクセス制御をする．図 5 にシステムの概要図を示す．

ユーザはユーザ名とパスワードを登録することにより，システムを利用できる．耐結託 ID はシステムの効率化のため事前にデータベースに登録しておく．システムは登録されたユーザのユーザ名と耐結託 ID を紐付ける．認証されたユーザは文書のアップロード，ダウンロードが可能となる．ユーザにアップロードされた文書は，タイトルや作成者といったメタデータと文書のハッシュ値とともにデータベースに保存される．文書ダウンロード時にはユーザの耐結託 ID を電子透かしで文書に埋込み，

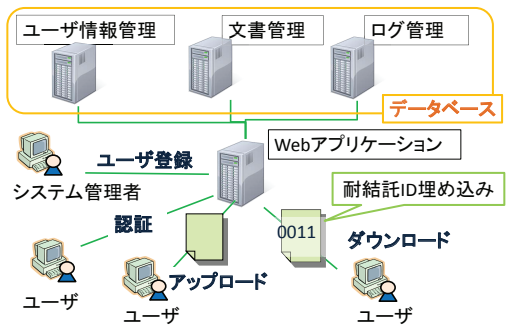

図 5: 耐結託 ID 埋込み機能を持つ電子文書管理システム

情報漏洩を抑止する．アップロードされた文書に耐結託 ID が埋込まれている場合は，耐結託 ID を削除して文書を保存する．

4.2　耐結託 ID 埋込み機能

テキストボックスを利用した電子透かしを用いて，本システムからダウンロードされる文書に耐結託 ID を埋込む機能を構成する．OOXML に理解のある者でなくては消去困難で，Word 2007 文書への文章のコピーで耐結託 ID が同時にコピーされやすい耐結託 ID 埋込み機能を実現する．

テキストボックスは一つテキストボックスに一つの耐結託 ID を書き込み，子要素に文字列を持つ`<w:p>`要素に埋込む．一つの文書に複数のテキストボックスを散在させることで，耐結託 ID がコピーされる可能性を高め削除される可能性を下げる．テキストボックスに書き込む耐結託 ID は大きさを 1pt とし，テキストボックス枠内収まるようにテキストボックスの大きさを調整する．

耐結託 ID の生成には Hou らが提案した ACC 符号を利用する．ACC 符号は文字列で埋込む場合は平均化攻撃が成立しないという問題があるため，符号アルファベットのランダム化を施す．この問題については次節で述べる．

4.3　符号アルファベットのランダム化

ACC 符号は平均化攻撃を想定した，0，1 で表される符号である．平均化攻撃では結託した符号の異なる部分をそれぞれ加算し，小数を含む平均値に改竄する．しかし，符号を文字列で文書に埋込む場合，符号アルファベットは任意に書き換えられる．この場合，結託攻撃による改竄で，符号のうち $p^3 - p$ 個以上のアルファベットが 1 となるように改竄されると結託した符号が特定不可能になる．

この問題に対処するため，符号アルファベットを一桁ごとにランダムに変更する．符号アルファベットのランダム化により，符号の 0，1 の判別を不可能にし，符号アルファベットを 1 に偏って改竄される可能性を低下させる．

ランダム化は Marking Assumption を成立させるため，同一の文書では同じ符号アルファベットを用いる必要がある．そこで符号埋込み前の文書のハッシュ値を元に符号アルファベットをランダム化する．符号アルファベットのランダム化の手順を以下に示す．

1. 符号を埋込む文書のハッシュ値を計算する．
2. 計算したハッシュ値を種にして乱数生成器 A を作成する．
3. 乱数生成器 A の乱数を種にして文書に埋込む符号の数 n と同数の乱数生成器 X_1〜X_n を作成

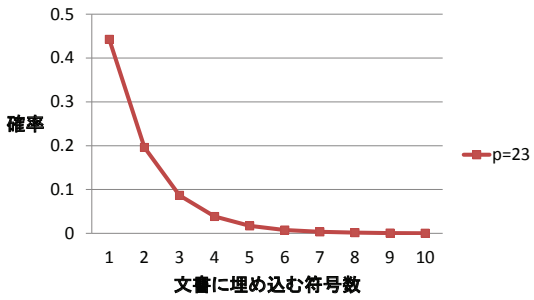

図 6: 結託者を一人も特定できない確率

する．

4. 乱数生成器 X から乱数を 2 つ取り出し，数を Base64 のキャラクタに変換する．
5. 変換したキャラクタを符号アルファベットの 0,1 と対応させ，符号の一桁を文字に置き換える．
6. 手順 4〜5 を符号の桁数と同じ回数繰り返す
7. 手順 6 を乱数生成器を変更して n 回繰り返す

文書に複数個の符号を埋込む場合には，符号アルファベットのランダム化に利用する乱数系列を符号ごとに変更し，符号ごとの差分から改竄箇所を特定可能にする．

4.4 結託者を一人も特定できない確率

符号アルファベットの擬似ランダム化を施すと，ACC 符号はランダム攻撃に対応できる．Hou らが提案した符号がランダム攻撃された場合の結託者が一人も特定できない確率を求める．

符号を作るための素数を p とし，符号の 0 の数を n_0，1 の数を n_1 とする．結託者が一人も特定できないのは $n_1 > p^3 - p$ のときである．表 1 より ACC 符号の符号長は $p^3 + 1$ なので，結託者が一人も特定できないとき n_0 は

$$n_0 < p + 1 \tag{1}$$

Marking Assumption の元では結託した符号の異なる部分しか改竄されない．ACC 符号は 0 の位置が符号ごとに別々であり，一つの符号の n_0 は $p+1$ 個である．したがって結託した符号を k 個とすると結託攻撃により改竄される符号の桁数は最大で $k \cdot (p+1)$ である．$k \cdot (p+1)$ 桁のアルファベットのうち 0 に改竄された桁数を s とすると，符号のうち s 桁が改竄される確率は

$$ {}_{k\cdot(p+1)}C_s \cdot 2^{-k\cdot(p+1)} \tag{2}$$

```
- <w:p>
  - <w:r>
    - <w:rPr>
        <w:sz w:val="1"/>
        <w:b/>
        <w:rStyle w:val="heading2"/>
        <w:lang w:val="en-CA"/>
      </w:rPr>
      <w:t>o</w:t>
    </w:r>
```

図 7: `document.xml` に埋込まれたランダム化した耐結託 ID の一桁

式 (2) は $k = 2$ のとき最大となる．したがって文書に埋込む符号数を x とすると，結託者が一人も特定できない確率の最大値は

$$\left(\int_0^p {}_{2\cdot(p+1)}C_s \cdot 2^{-2\cdot(p+1)} \cdot ds \right)^x \tag{3}$$

式 (3) は図 6 のようなグラフを描く．耐結託 ID で埋込む文書に符号数を 6〜10 個程度とすることで，結託者が一人も特定できない確率を 1 %以下に抑えられる．

4.5 要素ごとの結託攻撃への対応

耐結託 ID は 10^4 以上の符号長を持つことが想定される．耐結託 ID をそのままテキストボックスに書き込んだ場合，`<w:t>`要素ごと交換することにより，Marking Assumption が成立しなくなる危険性や，OOXML を眺めるだけで符号の埋込み箇所が特定される危険性がある．そこで，テキストボックスに埋込む耐結託 ID は 1 桁ごとに`<w:r>`要素で区切り，文字のフォントを示すような要素で修飾する．

図 7 に耐結託 ID を埋込んだ `document.xml` の一部を示す．図中下線部`<w:t>`要素の内容がランダム化した耐結託 ID 一桁を示している．図 7 のような埋込み方をすることにより，要素ごとに交換するような結託攻撃に対処し，目視による符号埋込み箇所特定を困難にする．

4.6 提案システムの情報漏洩抑止

提案システムからダウンロードした文書には，ダウンロードしたユーザの耐結託 ID が埋込まれる．文書に埋込まれた耐結託 ID の存在を事前に周知しておくことにより，正当なアクセス権を持つ者による結託攻撃を抑止することが可能になる．

5 提案システムの結託耐性評価

5.1 実験趣旨と評価項目

本章では，提案システムを組織で利用する場合に結託攻撃が起こりうるか，結託攻撃された文書の耐結託 ID

から結託したユーザを特定可能かを検証する実験をし，システムの実現可能性を評価する．

実験では異なる ID を持つユーザに提案システムから文書をダウンロードさせ，文書に埋込まれた耐結託 ID の改竄方法を検討させる．実験では以下の項目を評価する．

項目 1. ユーザが結託攻撃に必要なリテラシを備えているか

項目 2. ユーザが電子透かしを消去せずに結託攻撃するか

項目 3. 結託攻撃された文書の耐結託 ID から結託したユーザが特定できるか

5.2 実験概要

PC を 2 台用意し，提案システムにアクセス可能なユーザ ID，パスワードを二人一組の被験者にそれぞれ与える．被験者にそれぞれのユーザ ID で本システムにアクセスさせ，ダウンロードした文書に埋込まれた耐結託 ID の改竄方法の検討，文書に対する結託攻撃を実施させる．

実験を実施するにあたって被験者の行動に制限を設けた，改竄方法の検討では他のテキストエディタへの文書のコピー，文書画像のキャプチャによる ID の消去は禁止とした．また，Web ページの閲覧を禁止とした．代わりとして，改竄方法の検討では必要と思われる情報を経過時間ごとに与えた．

PC の操作に比較的習熟している者として，京都大学情報学研究科の学生 12 名を対象とした．

5.3 実験手順

実験手順は以下の通り，実験時間は合計 120 分とする．

1. 実験内容説明
2. 耐結託 ID の改竄方法検討（制限時間 60 分）
3. 検討した改竄方法の記入
4. 結託攻撃による耐結託 ID 改竄
5. リテラシに関するアンケート

耐結託 ID の改竄方法検討では，表 2 のように経過時間ごとにヒントとなる資料を与える．OOXML 文書の展開・圧縮方法の資料では，zip ファイルである OOXML 文書の展開・圧縮方法を図示する．代表的な電子透かし手法の資料では，画像の輝度情報を改変する手法，文書の改行を利用する手法 [5]，文書中の単語を類義語へ置き換える手法 [6] を示す．電子透かしでの攻撃手法の資料では，コンテンツにノイズを加える手法，圧縮等のコンテンツの形式を変換する手法，複数コンテンツを比較して差分を書き換える手法を示す．

結託攻撃による耐結託 ID 改竄では，結託攻撃の方法を実験者が説明し，被験者に結託攻撃により文書の耐結託 ID を改竄させる．

表 2: 改竄方法検討におけるヒントとその提示時間

時間	ヒント内容
10 分	OOXML 文書の展開・圧縮方法
30 分	代表的な電子透かし手法
50 分	電子透かしへの攻撃手法

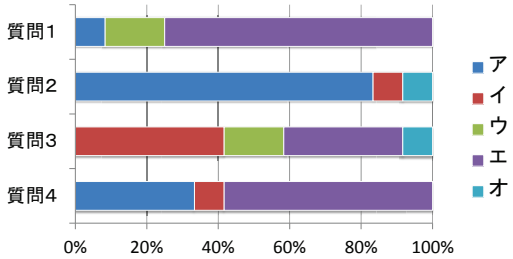

図 8: アンケート結果

5.4 実験結果

5.4.1 評価項目 1：ユーザが備えるリテラシについて

リテラシに関するアンケートの質問項目を付録 A に示す．アンケート結果は図 8 のようになった．

図 8 より，docx ファイルが zip 圧縮されたファイルであるという知識が被験者に最も不足していることがわかる．しかし，”docx ”という単語を Google 検索を利用して検索すると，docx ファイルが zip 圧縮されたファイルであることが記載された Web ページが上位に現れる[4]．したがってこの知識は Web の閲覧が可能であれば比較的容易に得られる．

一方，結託攻撃による耐結託 ID 改竄において改竄文書の作成に成功した被験者は 12 名中 5 名であった．残り 6 名は Word 2007 での表示に失敗する文書を作成した．文書作成に失敗した理由は，不正な文字の利用，バックアップファイルの消し忘れ，操作ミスによる不正な XML を作成であった．

こうした理由で文書作成に失敗した原因としてはバイナリエディタを利用したことが挙げられる．バイナリエディタは一般的なテキストエディタと文書の編集方法が異なる，変更した値が不正な文字であるかがユーザにはわからないという問題がある．また実験で利用したバイナリエディタは自動でバックアップファイルを作成するため，その存在を忘れやすい．

結託攻撃による耐結託 ID 改竄の結果からは，結託攻撃に必要なツールに対するリテラシが不足しているとい

[4] 2011 年 12 月 26 日の検索結果

える．しかし，ユーザが利用するツールに習熟するための時間があれば結託攻撃は可能であると推察できる．

5.4.2 評価項目 2：結託攻撃の実施可能性

耐結託 ID の改竄方法の検討において，12 名中，8 名が改竄方法として結託攻撃を挙げた．結託攻撃以外に検討された方法としては Word 2007 を利用して docx ファイルを doc ファイルに変換する，ダウンロードした文書と内容が同一の文書を新規作成し比較する等の方法があった．`document.xml` の電子透かし部分に情報が埋め込まれていることを疑う被験者は多かったが，検討した改竄方法として電子透かしの消去を解答した被験者はいなかった．以上より，ユーザが電子透かしを消去せずに結託攻撃する可能性は十分考えられる．

5.4.3 評価項目 3：結託耐性について

結託攻撃による耐結託 ID 改竄において被験者が作成した改竄文書の耐結託 ID から，結託したユーザの検出を試みた．Word 2007 で表示不可能な文書は表示可能となるように修正した．12 名中 11 名の文書の耐結託 ID から結託した者 1 人が特定可能であった．結託した者が特定出来なかった文書は Marking Assumption に反する編集がなされていた．また，12 名中 4 名の文書の耐結託 ID から結託した者 2 名が特定可能であった．

5.5 提案システムの実現可能性

実験結果より，正当なアクセス権を持つ者が，提案システムから取得した文書に対して結託攻撃した場合，結託したユーザを少なくとも一人特定できることが検証できた．さらに，ユーザは文書の耐結託 ID 改竄のための調査やツールに習熟する十分な時間があれば，文書に対して結託攻撃できるリテラシを備えている事がわかった．提案システムは，正当なアクセス権を持つ者による結託攻撃という実施される可能性のある問題に対して，情報漏洩抑止効果を持つ有用なシステムであるといえる．

6 おわりに

結託耐性符号で符号化したユーザ ID を利用することにより，情報漏洩抑止効果を高めたユーザ ID 埋込み機能を備えた電子文書管理システムを提案し，実現可能性を評価した．テキストボックスを利用した電子透かしを利用することで，文書の内容によらず大規模な組織でも利用できる ID 数を備えた耐結託 ID の埋込みを実現した．実現可能性の評価では，結託攻撃により改竄された文書の耐結託 ID から結託したユーザが少なくとも 1 人特定可能であること，正当なアクセス権を持つ者による結託攻撃が実施される可能性があることを示した．

A アンケートの質問項目

質問 1. docx ファイルが zip ファイルであることを知っているか
- **ア** 展開してみたことがある
- **イ** 知っている
- **ウ** 聞いたことがある
- **エ** 知らない

質問 2. zip ファイルの展開・圧縮をしたことがあるか
- **ア** 展開も圧縮もしたことがある
- **イ** どちらかは実施したことがある
- **ウ** どちらの実施方法も知っている
- **エ** どちらかの実施方法を知っている
- **オ** 実施方法がわからない

質問 3. 「電子透かし」を知っていたか
- **ア** よく知っている
- **イ** 簡単な方式が一つ思い浮かぶ
- **ウ** 言葉の意味を知っている
- **エ** 聞いたことがある
- **オ** 知らない

質問 4. テキスト比較ツールの存在を知っていたか
- **ア** 利用したことがある
- **イ** そのような機能を持つツールを利用したことがある
- **ウ** 存在は知っている
- **エ** あるだろうという予測はつく
- **オ** 知らない

参考文献

[1] NPO 日本ネットワークセキュリティ協会．2010 年情報セキュリティインシデントに関する調査報告書〜個人情報漏えい編〜, 2011.

[2] D. Boneh. Collusion-secure fingerprinting for digital data. *IEEE Trans. Information Theory*, Vol. 44, No. 5, pp. 1897 –1905, 1998.

[3] W. Trappe. Anti-collusion fingerprinting for multimedia. *IEEE Trans. Signal Processing*, Vol. 51, pp. 1069–1087, 2003.

[4] S Hou, T Uehara, T Satoh, Y Morimura, and M Minoh. Fingerprinting codes for internet-based live pay-tv system using balanced incomplete block designs. *IEICE Transactions on Information and Systems*, Vol. 92, No. 5, pp. 876–887, 2009.

[5] 滝澤修, 松本勉, 中川裕志. 改行位置の調整によるドキュメントへの情報ハイディング (情報セキュリティ特集) – (情報漏えい対策技術). 情報通信研究機構季報, Vol. 51, No. 1, pp. 153–169, 2005.

[6] 中川裕志, 木村浩康, 三瓶光司, 松本勉. 辞書変換法に基づく日本語テキストへの情報ハイディング. 情報処理学会論文誌, Vol. 41, No. 8, pp. 2272–2280, 2000.

[7] 北野宗之, 増田英孝, 中川裕志. Word 2003 xml 文書への情報ハイディングシステム. 電子情報通信学会技術研究報告. ISEC, 情報セキュリティ, Vol. 105, No. 193, pp. 205–212, 2005.